



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification 6: --- G06F 17/30</p>	<p>A1</p>	<p>(11) International Publication Number: WO 99/08208 (43) International Publication Date: 18 February 1999 (18.02.99)</p>
<p>(21) International Application Number: PCT/NZ98/00121 (22) International Filing Date: 12 August 1998 (12.08.98) (30) Priority Data: 328538 12 August 1997 (12.08.97) NZ (71) Applicant (for all designated States except US): AORAKI CORPORATION LIMITED [NZ/NZ]; 17 Sir William Pickering Drive, Bishopdale, Christchurch (NZ). (72) Inventor; and (75) Inventor/Applicant (for US only): SEGARAN, Chendra [NZ/NZ]; 17 Sir William Pickering Drive, Bishopdale, Christchurch (NZ). (74) Agents: CALHOUN, Douglas, C. et al.; A.J. Park & Son, Huddart Parker Building, 6th floor, Post Office Square, P.O. Box 949, Wellington 6015 (NZ).</p>		<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published With international search report.</p>
<p>(54) Title: INTERNET TRANSACTION PROCESSING INTERFACE</p> <div data-bbox="568 1134 1234 1701"> </div> <p>(57) Abstract</p> <p>An Internet transaction process interface for software applications residing in an object oriented client server environment which dynamically generates HTML code. Internet user requests are passed from a web server to the object oriented environment. User session management, manages and maintains web connected users via use of TCP/IP address lists, unique session IDs and their corresponding session objects. Each session object calls upon pre-stored application forms which includes menus, controls and implicitly assigned data from the user requests. The controls are executed via a method and the output returned to the form, this is then converted to corresponding HTML strings and passed back to the web browser.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

"INTERNET TRANSACTION PROCESSING INTERFACE"

TECHNICAL FIELD

This invention relates to an internet transaction processing interface.

5

BACKGROUND ART

Figure 1 represents the conventional manner in which a software application is "web enabled". By 'web enabled' is meant the provision of access via the world wide web to an application and associated database instead of via a private local area or wide
10 area network.

The web interface or "adapter" program 1 is usually developed as a totally separate computer program from the computer program which implements the in-house business transactions. Numerous products/tools are available that allow for the
15 development of both static and dynamic web pages. In this diagram the database is identified as 2 and the business processing program is identified as 3. The business processing program is connected to the enterprise LAN/WAN 4. In such prior art systems the web transactions are treated separately and differently from the standard business transactions. The web transactions are processed by a different program
20 which will be written in a different language using a different tool. Any significant change in the database structure would result in changes having to be made in both the web interface program and the business processing application. Additionally any significant change in the handling of any specific transaction would also need both the interface program and the business application program to be modified.

25

In many cases the traditional or typical environment available today is a mixture of a number of tools of different vendors. In the example illustrated in Figure 1 it would be likely that the web interface program, the database and the business program are all generated from totally different products potentially sourced from different
30 vendors. For example the database might be Microsoft SQL Server, the business program might be written in Visual Basic and the wen interface written in Perl.

DISCLOSURE OF INVENTION

It is an object of the present invention to provide an internet transaction processing interface which overcomes the above identified disadvantages of the prior.

- 5 Accordingly in one aspect the invention consists in an Internet transaction processing interface for software applications residing in an object oriented client server environment comprising
- (a) means for passing Internet user requests from a web sever to said object oriented environment,
 - 10 (b) means which manage web connected user sessions including maintaining a list of connected web user TCP/IP addresses together with unique system generated session IDs, checking requests for session IDs contained in said list and retrieving a corresponding session object if found, while if not found creating a new session object,
 - 15 (c) means which accept requests from each session object to call an appropriate pre-stored application form including menus and controls and assigns data implicit in said requests to said controls,
 - (d) means which cause a method associated with said controls to be executed and to return output data to said form,
 - 20 (e) means which accept said form, controls and menus and generates an HTML string by substituting html commands for corresponding form, controls and menus said HTML string capable of displaying on a web browser a page which corresponds to said form, said request passing means receiving said HTML string and passing it to the web server.

25

In a second aspect the invention consists in a method of providing Internet access to software applications residing in an object oriented client server environment comprising

- receiving Internet user requests at a web server, passing Internet user requests
- 30 from said web server to said object oriented environment,
- producing for each new session a unique system generated session ID,

- 3 -

maintaining a list of connected web user TCP/IP addresses, together with said unique system generated session IDs, checking requests for session IDs contained in said list and retrieving a corresponding session object if found, while if not found creating a new session object,

5 accepting requests from each session object to call an appropriate pre-stored application form including menus and controls, assigning data implicit in said requests to said controls, executing a method associated with said controls, said method returning output data to said form,

10 generating an HTML string by substituting corresponding html commands for said form, controls and menus, said HTML string capable of displaying on a web browser a page which corresponds to said form, and passing said HTML string to the web server.

15 In the present invention the web interface, the business processing and the database are all integrated and operating in real time and are sourced from the same development environment. The essential processing steps performed by the interface are shown in figure 4, where the IIS (Internet Information Server) is standard web server software, such as Microsoft Internet Information Server.

20 The advantages of the preferred form of the present invention include increased reliability as only a single development tool is involved, reduced costs of operation as only a single entity needs to be maintained, consistency of view and that all data is potentially available to all users, consistency of processing as the code processing

25 a web initiated transaction can be the same code used for processing a LAN/WAN transaction.

Strictly speaking, all web pages are dynamically create although the facility to include static pages as a part of dynamic generate is possible. A developer can

30 create an application in one complete graphical environment and deploy this application on the web.

BRIEF DESCRIPTION OF THE DRAWINGS

in which

Figure 1 represents the existing typical internet transaction processing interface;

5 Figure 2 is the transactional processing interface according to the present invention broadly represented;

Figure 3 is a more detailed representation of the transactional processing interface according to the present invention;

10 Figure 4 is a flow diagram representing the sequence of processing events carried out in the present invention, and

Figure 5 is a diagram showing how the present invention serves as an interface to more than one software application.

BEST MODES FOR CARRYING OUT THE INVENTION

15 The present invention in one aspect is diagrammatically and simply represented in Figure 2. A single application established by the development environment 5 can receive input from and deliver output to both the web 6 and the standard LAN/WAN environment 7 of a business enterprise. Any change to the database (repository) or the program is instantly available to all potential users as the development
20 environment 5 ensures consistency.

In this way the web interface, the business processing application and database are all integrated and sourced from the same development environment, leading to the advantages as specified above.

25

The essential features of the web interface of the present invention will now be described

1. Automatic Generation of Web Interfaces

30

The same forms developed for an application deployed on a LAN/WAN are

- 5 -

made available for deployment on the web thereby allowing the application to be run from the web. HTML code is automatically generated for each form accessed by a web user.

5 Typically, the HTML code generated will conform to the HTML 3.2 standard. This allows a developer to generate a single user interface that is compatible with multiple browsers.

2. Session Management

10

A session is created for each user that accesses an application through the web. A session will store information such as the session ID, time created, form name and client address. When the application is first accessed a unique random session ID will be assigned to the user. This ID will be included on every form while remaining invisible to the user.

15

4. Application Clients

20

Application clients running on the LAN can also be running the same application as is being run from the web. The user interface forms for the LAN client can be the same as the ones defined for the web or they can be different. This allows the application developer to have one application with the same or different views of that application data as between LAN and web users.

25

In the preferred embodiment of the invention Microsoft Internet Information Server (IIS) is used as the web server. ISAPI is used to communicate with the IIS.

30

Security is implemented using Windows NT Security and IIS security for data access and secure sockets layer for data transmission. All these facilities are

provided by Windows NT and the Internet Information Server and is not specific to the JADE environment.

5 Figure 3 shows a more detailed view of the internet transaction processing interface according to the present invention. A web server PC 8 is shown web running an application 9 with a Repository 10 and a web interface 11 according to the present invention. A remote Internet connected PC 13, having a world wide web browser 14 displaying an HTML page 15 is connected to the server 8. At the same time the application 9 is connected
10 through the LAN/WAN 16 to a client PC 17 operating a client node 18 and application windows (forms) 19. The web client PC may be running an NT, Win.95, W3.11, Mac or Unix operating system and any HTML 3.2 compatible browser.

15 It should be appreciated that the present web interface allows access to more than one business application running on the server. This is illustrated in figure 5 where three applications 9a, 9b and 9c are shown by way of example. A web user can access any one of these applications provided the application name is included in the URL entered in the web browser. In addition many
20 web users may concurrently access the same business application.

The Internet transaction processing interface of the present invention is preferably incorporated in an integrated object oriented development and run-time environment such as JADE™ (produced by Aoraki Corporation Ltd and
25 described in the JADE Technical Overview (4.0), published in August 1997). A more detailed description of the invention will now be given with reference to JADE.

Automatic Generation of the Web Interface

30

Within the JADE development environment the JADE forms painter offers a

web page as an additional form style on the "create new form" dialogue.

When this is selected as the form style the painter switches to a mode that supports HTML style windows. The form creation grid will automatically be set up to support character-mode style painting.

5

A web insert control allows the contents of a file containing HTML code to be inserted as part of the HTML generation. The file contents are copied as is. The web HTML control allows text to be directly inserted into the generated HTML.

10

A WEBHTML control allows for the direct insertion of HTML code. This code is then included as a part of the dynamically generate HTML. There is also a hot spot control that allows hot spots to be placed on an image.

15

The HTML generation process starts when the "show" method (code segment) of the form is called. The generation is as follows: the form load method is invoked. This method must set up all the required information in the controls e.g. a combo box must be populated in this method. Once the load process has completed then the HTML generation is started by calling the output HTML method for the form, the menu items and the controls. This generation creates a string, containing HTML text. The string is then returned to the web browser which then displays the information. A range of controls that have an equivalent HTML 3.2 syntax are provided. Image generation is allowed for, as part of the HTML generation. When an image is encountered,

20

25

Examples of JADE painter - HTML syntax correspondence are:

Jade Form Class	HTML Equivalent
Form	<Form>

30

Jade Control Class	HTML Equivalent
Button	< INPUT TYPE = SUBMITS>

Jade Menu Class	HTML Equivalent
Menu Item	Generated as static text

2. Session Management

Whenever a new JADE schema is created a sub class of WebSchema is created for it. This allows the application developer to keep information relating to a web session. The schema dialogue allows the default web session sub class name to be changed. A new system property currentSession is available for use in the JADE logic. This property is used to get the session object for the current web session. The property will be null if there is no current web session.

A transient instance of WebSession is created when the application is started and stored in the application property in the WebSessionManager. When an internet request is received by the JADE application, the web sessions dictionary in the web session manager instance is accessed to determine whether the session is for an existing session. This is done by searching the dictionary using the TCP/IP address and the encrypted content of the hidden field "Session ID". If there is no such dictionary entry a new instance of the web session sub class is created. The TCP/IP address for the new user is set in the object (Client IP address). A new session ID is generated and set and the object is added to the web session dictionary.

The encrypted version of the Session ID is converted to a hexadecimal string and placed in the encrypted session ID property. This field is used as a key to the dictionary, rather than a session ID to save the overhead of encrypting the returned hidden field which would end up with the session ID.

If multiple copies of the application are initiated then they will use the same copy of the web session manager instance.

5 The encrypted session ID is embedded as a hidden text field in every web page that is sent to the user. Creation time stamp of the web page is also stored as a hidden text field on the page. The last access time stamp is used to determine disconnect status. If there is no activity for a session for the user specified time then the session is ended.

10 3. Web Applications

JADE applications can be web enabled by setting the web enabled check box. If an application is web enabled then the following will apply.

15 The startup form defined for the application is the first web page that the user sees when they invoke the application from the web browser. The startup form that is displayed when the user runs the application is a form that is provided with JADE. This form will allow monitoring and control the web requests. A web enabled application will also invoke its own initialize method
20 (webInitialise) and this cannot be overridden. Application features such as MDI and 3D controls are also ignored for web enabled application.

4. Web Forms

25 Except for the previously mentioned Painter facility JADE does not distinguish between web forms and non-web forms. However, depending on whether the application is web enabled or not the process involved in displaying the form will be different.

5. Event Handling

Click methods on controls will be executed as at present. If a click method is present the generated HTML has the URL set up and a click on this control will cause a request to be sent back to the JADE application. No other event method will be processed (unless it is explicitly called from the click method). When a request is sent to the JADE application the web browser waits for a reply. If the JADE code that is processed does not initiate the creation of another web page then the form that was previously displayed is redisplayed. If there were no previous forms then the start up form is displayed. This is so the web browser does not end up waiting forever for a response.

6. Unhandled Exceptions

The web application is armed with a global exception handler. When an unhandled exception occurs the error is recorded to the web user and logged on the JADE client computer. The JADE client is also informed of this error, the operation aborted and the web user is returned to the previous page.

7. Message Box Handling

The MSG box method in the application class is reimplemented in the root schema application class. This creates a HTML page and returns to the web user. As there is no modal capabilities with web applications any code following an application message box will continue to be executed.

8. Web Browser

The HTML code that is generated by JADE conforms to the HTML 3.2 specification. Specific Netscape or Explorer extensions are not used. The

user can still add these specifics by inserting raw HTML into the web page as previously described.

9. The JADE Web Connectivity

5

The processing of requests from a Microsoft Internet Information Server, (IIS) is handled by the following components (see also figure 4):

10 A The web interface software, jadehttp.dll, which is directly called by IIS for each internet request made. This dll will connect to a JADE system application via a named pipe (see B below). The jadehttp.dll sends the received user request over the channel to the JADE application, which processes the request and returns an HTML page for transmission to the internet user.

15 B A JADE class *InternetPipe* provides the ability for a JADE system to establish a communications channel with another program. To communicate with the jadehttp.dll, the application creates an instance of this class and then offers the named pipe for opening. Once the dll opens the other end of the channel, the JADE system waits for input from the dll. When input arrives, the JADE system processes that input and sends the reply back to the dll. The JADE system waits for more input.

25 Because the jadehttp.dll communicates directly with a standard JADE application, there are no limitations placed on the JADE facilities that can be used.

The JADE system asynchronously waits for input from the named pipe. This means that the JADE client is able to perform other tasks when idle such as monitoring and display of the system processing status.

30 The jadehttp.dll has been generalised so that communication can be initiated with any JADE system.

10. Initiation Flow

Each schema in the JADE database can be made web capable by defining a web-enabled application instance. This JADE application then communicates with the jadehttp.dll when it is initiated. This application opens an InternetPipe class instance with the name of the application as its name and waits for the jadehttp.dll to connect to the other end of the pipe. Once the pipe is connected, it waits for Internet requests to be sent over the pipe.

When the first request for the JADE application is received, the IIS initiates the jadehttp.dll and calls the GetExtensionVersion entry point in the dll to obtain the ISAPI version being used. The dll initialises itself as part of this call and attempts to open a pipe using the application name that must be passed as part of the initiation. It does not matter at this stage whether the open was successful or not. The dll assumes that the pipe is to be opened on the same device unless it finds a .ini file named jadehttp.ini with a JadeHost entry in a section with the application name.

To process multiple Internet requests simultaneously, additional copies of the JADE application can be run. Each copy opens its own instance of the named pipe and the jadehttp.dll will utilise those additional pipe channels as required.

11. Process Flow

25

The process flow for the handling of the JADE application will be as follows (see also figure 5):

I. The Internet Client will connect to the JADE application by invoking the jadehttp.dll passing the application name in the script directive.

30

II. This directive causes the IIS server to pass requests to this dll.

III. When there is such a script directive, the IIS call HttpExtensionProc entry point in the jadehttp.dll, passing the information describing the client request.

IV. The HttpExtensionProc routine will obtain and lock an available named pipe connection and send the request to the attached JADE application process. It then waits for the reply from the JADE system, which when received is sent to the internet client and the name pipe connection is unlocked. If there are other queued internet requests waiting (in a 'first in first out' queue), the first queued entry is removed from the queue and its associated thread awakened (see VII below). The thread finally exits back to the IIS caller.

V. If no named pipe channels are open, the open is attempted. If still no channel is available, the internet user will be sent the contents of the file 'jadehttp.htm' which informs them that the service is not available. If that file is not available, a simple 'service not available' message will be sent. If a channel was opened, the process restarts at IV above.

VI. If there are named pipe channels open but all are in use, every tenth time this occurs, an attempt will be made to open another pipe connection. For this to succeed, another copy of the JADE application must have been started. If the open succeeds, the processing continues as at IV.

VII. If there are still no available connections, the request will be added to a first in first out queue and the thread halts waiting for its turn at the available pipe channels. These entries will be processed when another thread completes its processing as at step IV.

VIII. If a pipe channel is broken by the connected JADE application terminating, the dll will remove that pipe from the list of available channels. When the JADE application is restarted, because of the above scheme, the dll will re-establish the channel transparently without any intervention required.

The HttpExtensionProc entry point can be called simultaneously by multiple IIS threads, and so the dll is 'thread safe'. This means each thread is able to execute simultaneously without affecting other thread copies.

Certain sections of the logic must be singled threaded to ensure integrity. For instance, locking an available pipe instance. This is achieved by using the Window's standard Critical Section facility. The single threaded sections of logic are kept to an absolute minimum.

12. Jade Processing

When the JADE system receives an internet request, the processing is as follows (see also figure 4):

- I. From the input data, the TCP address and the contents of a hidden field containing the encrypted session id are obtained.
- II. These fields are used to search the session dictionary for an existing session object for this client. If the entry is not found in the dictionary, a new client session object is created (See Client Session Control below).
- III. A method 'processRequest' of the session object is then called, passing the requested data to the system. This method must process the request and ultimately send a formatted HTML page back via the

named pipe instance on which the input data was received. The jadehttp.dll sends this page to the user and the IIS server is informed that the request processing is complete.

5 13. Termination Flow

If the web service of the IIS is stopped, it calls the TerminateExtension entry point of the jadehttp.dll. This function closes the named pipe connections to the JADE application.

10

The JADE application threads treat the pipe closures as an instruction to terminate.

14. Client Session Control

15

Client Internet sessions are managed as follows:

20 I. A list (dictionary) of active client internet sessions is maintained in the JADE database. This list is accessed using the client's TCP/IP address and a hidden encrypted session ID transmitted and returned with each HTML page for that client. The TCP/IP address is not necessarily unique across users, but the session ID is. The use of the TCP/IP address provides an additional level of security protection. Encrypting the session ID minimises the possibility of another user forging access, as both the TCP/IP address and the encrypted session key must be
25 valid while the other user is logged on.

30 II. The encryption is achieved using the standard Microsoft RSA encryption APIs. Each time the JADE application is initiated, a new 40 bit encryption key will be generated for the encryption process. The 4 byte session ID will be encrypted into a 64 bit key using the

standard block encryption method available.

- 5 III. When a user request is received, the request data is scanned for the hidden session ID information. If not present, or the TCP/IP and decrypted session ID combination is not located in the session list, a new session is commenced. This consists of creating a new unique session ID and a client session object in the JADE database and adding an entry to the session list for the TCP/IP address and session ID.
- 10 IV. If the TCP/IP and decrypted session ID combination is found in the session list, then the existing associated session object (information) is retrieved.
- 15 V. Each request received from that client session will be passed to that same session object. This is achieved by calling the 'processRequest' method of that session object.
- 20 VI. The JADE system will detect a new client session by the absence of a session object for that user.
- 25 VII. In this case, a new session object will created for that client.
- VIII. These session objects will be "timed out" after an application defined time. If no requests are received within that time limit, then the session will be deemed to have "timed out". Should the user transmit a request after that "timed out" period, they will be provided with a "timed out" message.
- 30 XI. Session objects which have "timed out" will be deleted after a defined time. This will have the same effect as in (VIII) above except that no "timed out" explanation will be given.

- X. Should the JADE application be taken down or crash, no session will be deemed to remain open. Users will be forced to reconnect and logon again.

5 15. JADE Named Pipe Class

The JADE application uses a subclass of NamedPipe to define the requirements for handling the pipe channel to the jadehttp.dll.

- 10 When the application is initiated, it creates an instance of the InternetNamedPipe which is stored in the WebSessionManager property internetPipe. An asynchronous open is then requested on that pipe.

- 15 When the jadehttp.dll opens the other end of the pipe, the openPipeCallback method is called. This method initiates an asynchronous read on the opened pipe.

- 20 When there is data available on the pipe, the readPipeCallback method is called. This method performs the session evaluation processing as described previously. The processRequest method is called for the internet session object created or found.

- 25 When the processing of the request is completed, the formatted HTML page will be sent back to the jadehttp.dll by calling the reply method of the session object. This method sends the reply to the jadehttp.dll, deletes the session object if required and initiates another read on the named pipe.

16. Logging

- 30 Should a problem occur with any processing in the jadehttp.dll, an entry will be posted to a file named 'jadehttp.log' located in the same directory as the

dll. Any failures that occur will be as a result of the JADE system's unavailability.

5 The IIS server will log all Internet requests received into its own log file. IIS logging should be enabled and it is recommended that the option for a new file to be automatically started each day be set.

10 The jadehttp.dll will also post routine messages into the log when initiated, closed and when connections to the JADE Named Pipes channels are made and broken.

17. Ini File

15 When the jadehttp.dll is initiated, it will search for a .ini file named 'jadehttp.ini' file located in the same directory as the jadehttp.dll:

20 If the file is not found, or there is no section with the application name or there is no JadeHost parameter, then it is assumed that the JADE System is located on the same device as the IIS and '.' is assumed.

18. Web Server

25 Microsoft Internet Information Server 2.0 is supported. The IIS has to be set up with virtual directories and the appropriate paths defined. As this is an IIS function, it will not be described here.

19. Security

30 Security can be defined at several levels.

I. Windows NT Security

Windows NT security will protect the computer and its resources by requiring assigned user accounts and passwords. It can control access to computer resources by limiting the user rights of these accounts. It can use the Windows NT File System (NTFS) to assign permissions to folders and files on the computer. You can control access to folders and files by preventing users from copying files to or from a folder, or by preventing users from executing files in certain folders.

II. Internet Information Server Security

In addition to Windows NT security, IIS provides additional security by restricting IP addresses, or checking user names and passwords (basic or Windows NT Challenge/Response). In addition, it also provides a security protocol, called Secure Sockets Layer (SSL). This protocol is layered between its service protocols (HTTP) and TCP/IP. The SSL provides data encryption, server authentication and message integrity for a TCP/IP connection.

III. Application Security

The user application can define its own security scheme via the facilities available within JADE.

20. User Preferences

The JADE application developer can set the following preferences:

I. showModal forms.

Display an error or treat the showModal command as a show. Default behaviour is to display an error.

II. Session timeout.

Specifies the timeout period that determines when the session is to end.
Default is 60 minutes. A value of 0 means no timeout.

5

Support for JAVA Applet

A JADE developer can "paint" a JAVA applet into the JADE painter to be included as part of the HTML generation. These JAVA applets are created outside of JADE.

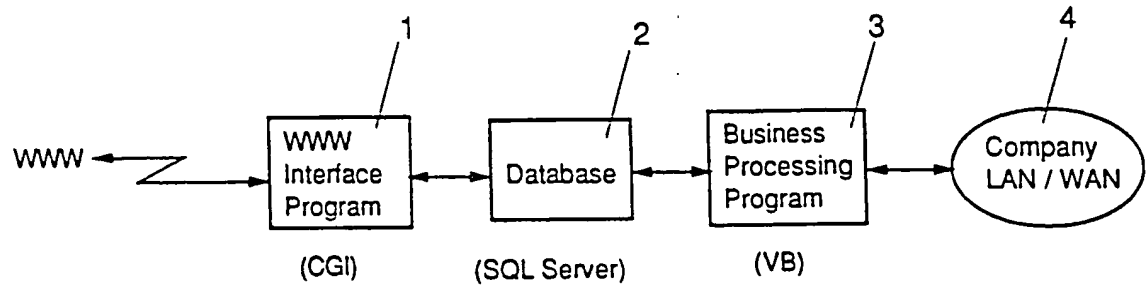
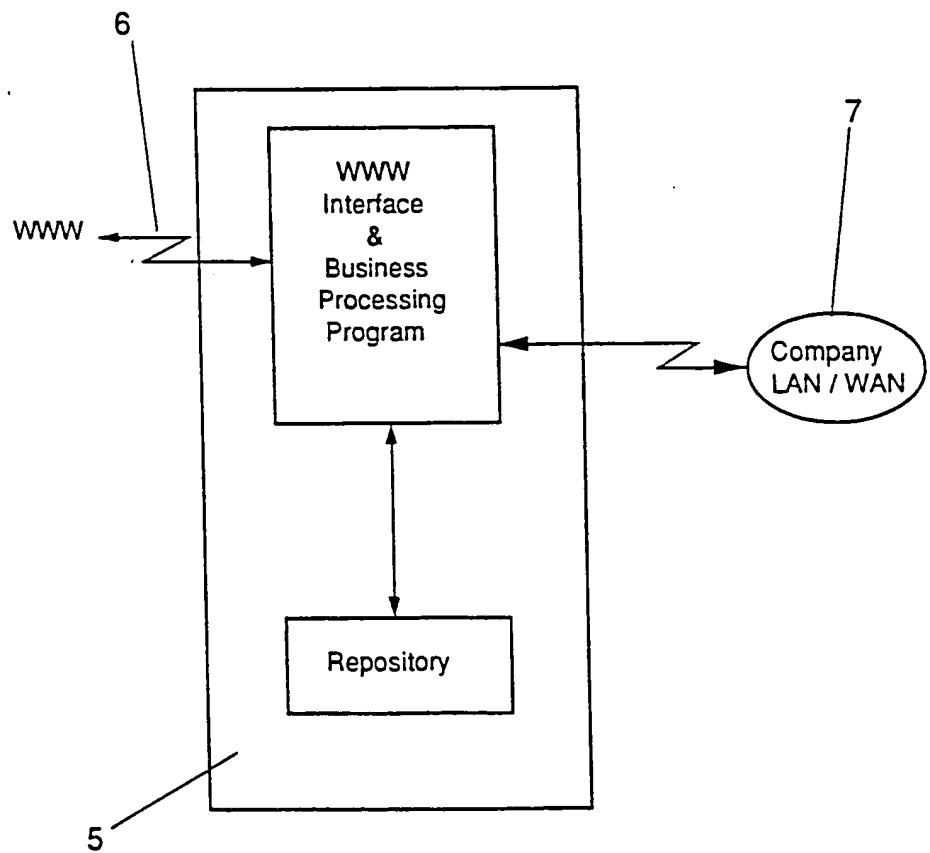
10

CLAIMS:

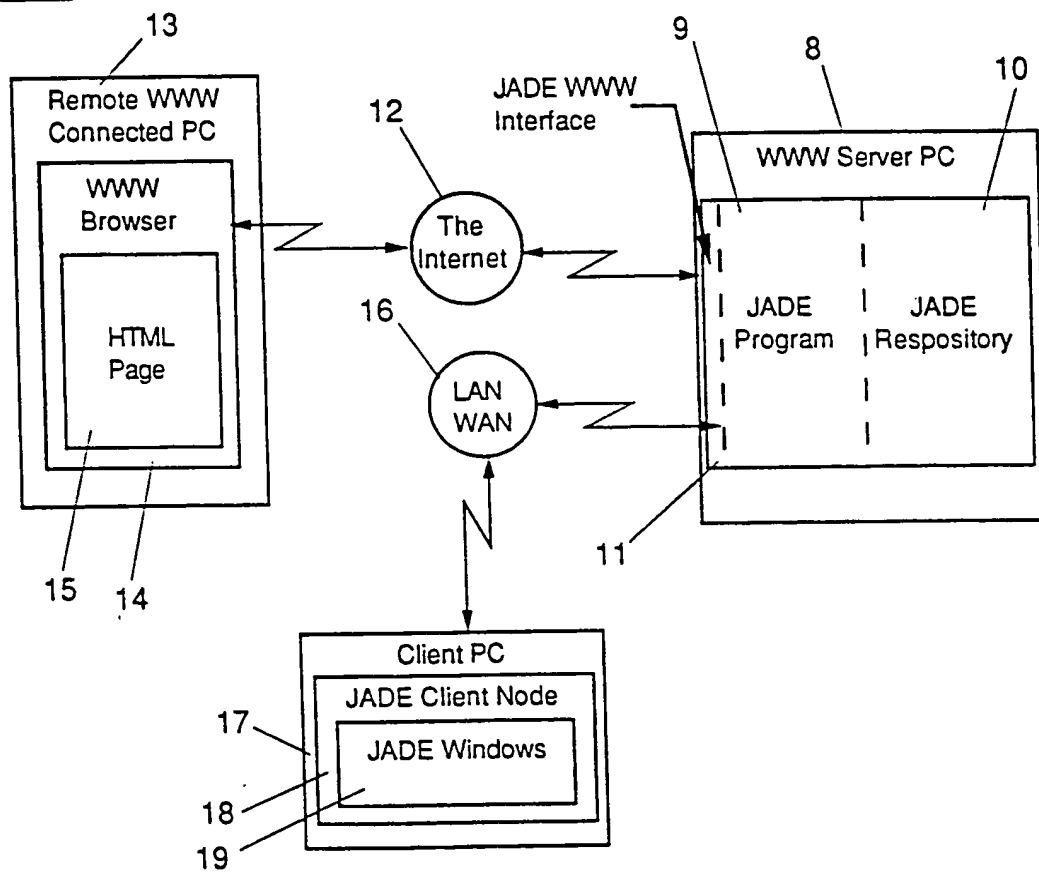
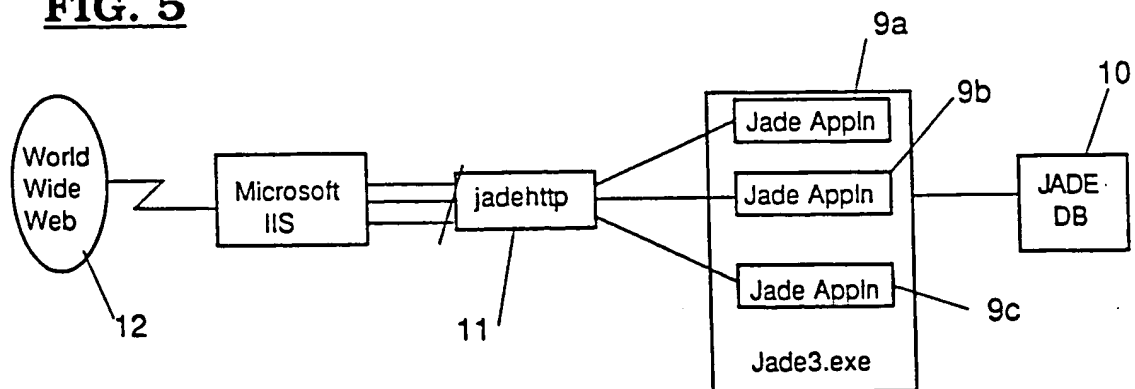
1. An Internet transaction processing interface for software applications residing in an object oriented client server environment comprising
 - a. means for passing Internet user requests from a web sever to said object oriented environment,
 - 5 b. means which manage web connected user sessions including maintaining a list of connected web user TCP/IP addresses together with unique system generated session IDs, checking requests for session IDs contained in said list and retrieving a corresponding session object if found, while if not found creating a new session object,
 - 10 c. means which accept requests from each session object to call an appropriate pre-stored application form including menus and controls and assigns data implicit in said requests to said controls,
 - d. means which cause a method associated with said controls to be executed and to return output data to said form,
 - 15 e. means which accept said form, controls and menus and generates an HTML string by substituting html commands for corresponding form, controls and menus said HTML string capable of displaying on a web browser a page which corresponds to said form, said request passing means receiving said HTML string and passing it to the web server.
 - 20
2. An Internet transaction processing interface according to claim 1 further including means which establish communication channels between said request passing means and any one of a plurality of applications residing in said client server environment, each channel corresponding to a different web user request.
- 25
3. A method of providing Internet access to software applications residing in an object oriented client server environment comprising
 - 30 receiving Internet user requests at a web server, passing Internet user requests from said web server to said object oriented environment,

- producing for each new session a unique system generated session ID,
maintaining a list of connected web user TCP/IP addresses, together with said
unique system generated session IDs, checking requests for session IDs
contained in said list and retrieving a corresponding session object if found,
5 while if not found creating a new session object,
accepting requests from each session object to call an appropriate pre-stored
application form including menus and controls,
assigning data implicit in said requests to said controls,
executing a method associated with said controls, said method returning
10 output data to said form,
generating an HTML string by substituting corresponding html commands for
said form, controls and menus, said HTML string capable of displaying on a
web browser a page which corresponds to said form, and passing said HTML
string to the web server.
- 15
4. A method according to claim 3 including the step establishing from the
Internet request the selected application from a plurality of applications
residing in said client server environment, and ensuring that for the duration
of the session all requests forming part of that session communicate with said
20 selected application.

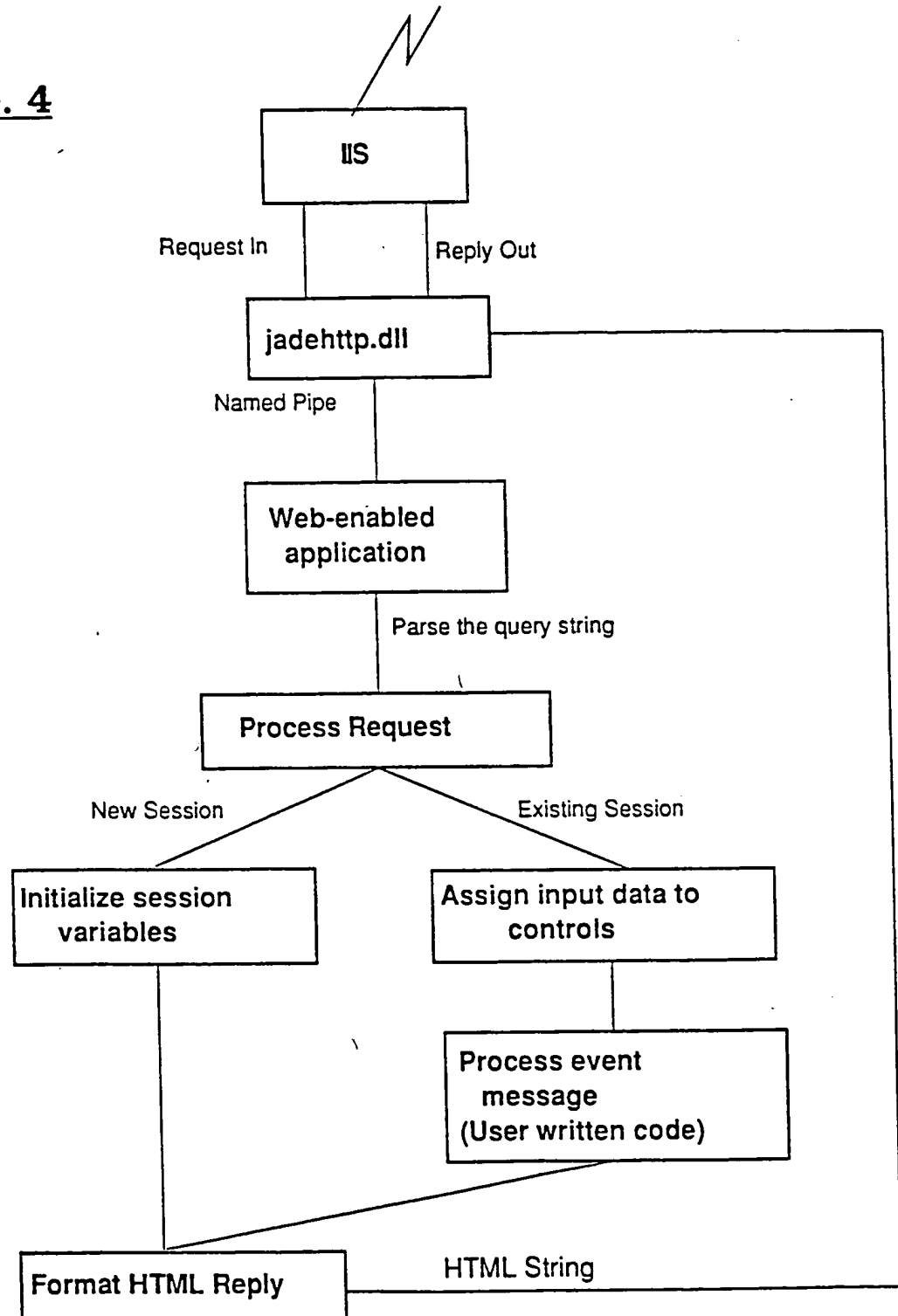
1/3

FIG. 1**FIG. 2**

2/3

FIG. 3**FIG. 5**

3/3

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/NZ 98/00121

A. CLASSIFICATION OF SUBJECT MATTER		
Int Cl ⁶ : G06F 17/30		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 17/30		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WPAT, BYTE, INTERNET, WEBTECHNIQUES.		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P. A	US 5754772 A (LEAF) 19 May 1998 Whole document Figure 2 Column 4 lines 50/67	1-4
P. A	US 5742762 A (SCHOLL et al) 21 April 1998 Whole document Column 6 line 37 to column 7 line 4	1-4
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C <input checked="" type="checkbox"/> See patent family annex		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search 6 November 1998		Date of mailing of the international search report 10 NOV 1998
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No.: (02) 6285 3929		Authorized officer R STOPFORD Telephone No.: (02) 6283 2177

INTERNATIONAL SEARCH REPORT

International application No.
PCT/NZ 98/00121

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, A	US 5737592 A (NGUYEN et al) 7 April 1998 Whole document Column 4 lines 54/62	1-4
A	EP 747845 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 11 December 1996 Whole document	1-4
A	BYTE MAGAZINE-September 1996. "The Web Is Changing the Way Companies Automate Forms Processing". Peter Jerram. HTTP://WWW.BYTE.COM/ART/9609/SEC11/ART2.HTM. Whole document	1-4
A	WEB TECHNIQUES MAGAZINE-July 1997. "A Design Retrospect From One Of Web Objects' top developers". Bob Kachms. HTTP://WWW.WEBTECHNIQUES.COM/FEATURES/1997/07/KAEHMS/KAEHMS.HTML Whole document	1-4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/NZ 98/00121

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report				Patent Family Member			
EP	747845	CA	2177917	CN	1138177	JP	9026972
		US	5721908				
END OF ANNEX							